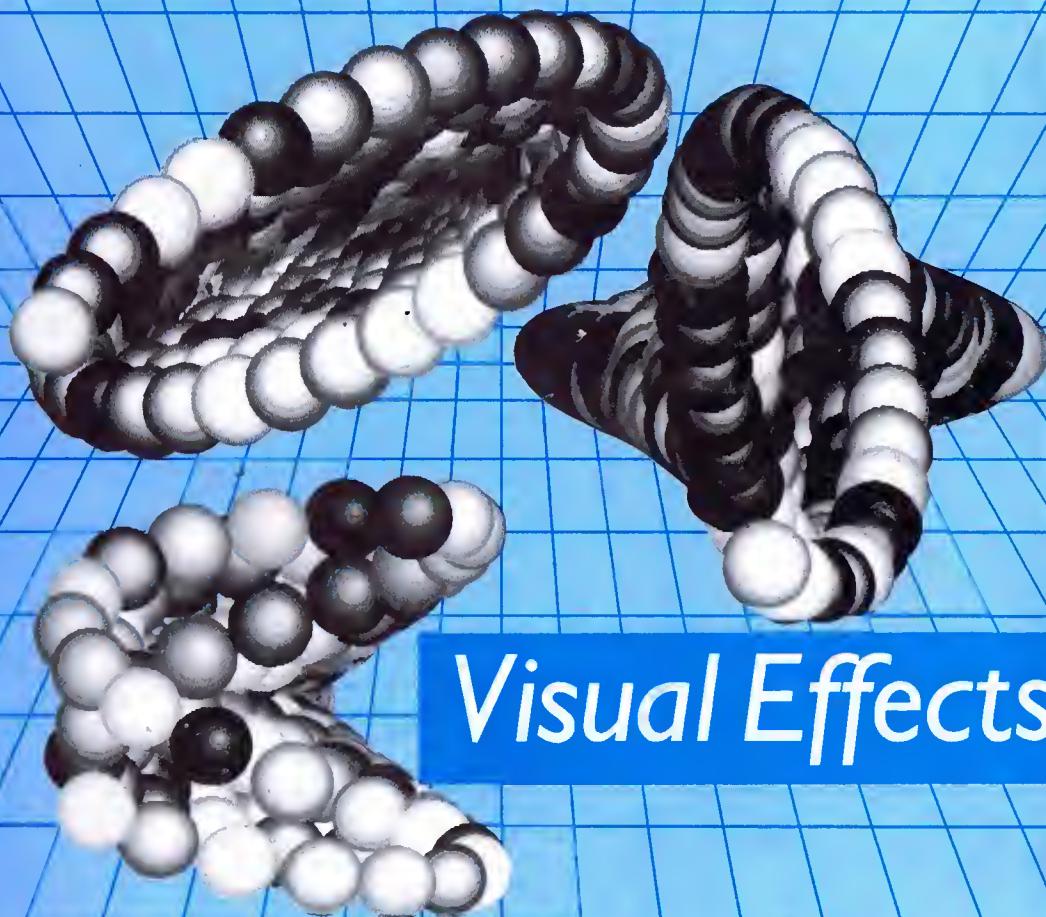
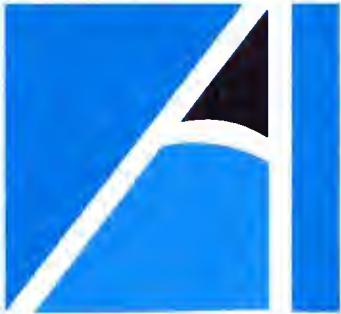


Volume 1 Issue 1

November  
1987

Price £1.20

# RISC USER



## Visual Effects

THE MAGAZINE AND SUPPORT GROUP  
EXCLUSIVELY FOR USERS OF THE ARCHIMEDES

# RISC USER

## CONTENTS

<b>Editorial</b>	3
<b>News and Comment</b>	4
<b>Archimedes Disc Menu</b>	6
An automated ADFS menu for Archle	
<b>Archie's White Mouse</b>	10
Putting the mouse to work	
<b>Painting with Archimedes</b>	14
Welcome Paint and Artisan compared	
<b>Sound and Music</b>	17
Getting started with SOUND and VOICE	
<b>Using Discs</b>	19
Understanding the disc filing system	
<b>Visual Effects</b>	22
Clever graphics techniques	
<b>Basic Editor and Emulator</b>	24
Making the most of these utilities	
<b>An Alias for Arthur</b>	25
Experimenting with new O. S. commands	
<b>Off the Shelf</b>	27
The first products for Archimedes reviewed	
<b>Hints &amp; Tips</b>	30

RISC User is published by BEEBUG Ltd.  
Co-Editors: Mike Williams, Lee Calcraft  
Assistant Editor: Kristina Lucas  
Production Assistant: Yolanda Turuelo  
Subscriptions: Sara Taylor

BEEBUG, Dolphin Place, Holywell Hill, St.Albans,  
Herts AL1 1EX. Tel. St.Albans (0727) 40303

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Limited.

BEEBUG Ltd (c) 1987



## The Archimedes Magazine and Support Group.

### EDITORIAL

Welcome to the first issue of RISC User, the magazine for all users of the Archimedes, Acorn's new computer system based on RISC technology. RISC User aims to provide in-depth practical information essential to all Archimedes users, programs which will provide readers with substantial utilities and applications for the new machine, as well as many demonstrations and examples of programming techniques. Reviews will also feature strongly, together with up-to-the-minute news on all Archimedes happenings. We have well established links with Acorn personnel, but we are independent and will be critical of all suppliers where appropriate. We also aim to cater for all users from beginner to expert, with a range of articles in each issue. In that way your copies of RISC User will build into an indispensable source of information on your Archimedes, one that you can refer to time and time again.

RISC User is not only a magazine. We aim to provide the best all-round support for Archimedes users, based on over five years experience with the BBC micro and Master series through our magazine and user group, BEEBUG. We offer you technical support by phone or by letter, all as part of your subscription, and discounts on what will be a growing range of new products for Archimedes, all in addition to a first rate magazine.

These are very early days for Archimedes, and we are all very much learning together how to derive the maximum benefit from this superb machine. We hope very much that you will write to us and tell us what you like and dislike about RISC User. We will also be only too delighted if you wish to share your experiences of Archimedes with other readers, and contribute information, articles and programs for publication. That way we will all learn, and we pay for all material used.

The Archimedes is a major achievement on the part of Acorn, and we certainly wish it every success. We shall be doing all we can to promote Archimedes and to help Archimedes users through RISC User. We hope you feel the same and will work with us in this endeavour.

#### PROGRAM NOTES

All the programs published in RISC User will be fully tested by our own staff, and listed out directly from working programs. However, the smaller memory on the A305, compared with the A310, will mean that the former system may need to be re-configured to suit the needs of different programs. For A305 users we will indicate where necessary the minimum values of SCREENSIZE, SPRITESIZE or RMASIZE needed. If no value is given then the defaults on your system will be adequate.

Lee Calcraft, Mike Williams

# News News News News News

The following report has been compiled to provide the latest information on software and other products for Archimedes. Some products available now have been reviewed in "Off The Shelf" in this issue. The information here is essentially news items on likely releases from various suppliers and software houses.

All prices quoted below include VAT, and are followed by the expected release date.

## ACORN

Without any warning, Acorn announced in September that there would be a reduction of £115 (inc. VAT) on all 300 Series machines as from 10th October. Acorn has also announced the A310M, which incorporates the PC emulator, at an extra £69 (inc. VAT) on top of the A310 price.

**SOFTWARE APPLICATION CATALOGUE** is now into its 3rd edition and is 72 pages long. We can supply this on receipt of an A5 SAE. 0% FINANCE. Acorn has set up a scheme whereby purchasers of full-price Archimedes 310 systems may, for as little as 1/12th down-payment, pay the remainder over the following twelve months. Payments being interest free. The scheme will only operate until 31st December 1987 and is available through all registered Archimedes dealers (including BEEBUG). However, it will take 3-4 days to arrange. Phone us for details.

**£50 EDUCATIONAL VOUCHERS.** Whenever an educational establishment buys an Archimedes it will receive a £50 voucher, which when sent to Acorn will be returned as a £50 cheque.

**I/O PODULES.** Code: AKA1, £90.85. In stock NOW.

**ROM PODULE.** Code: AKA0, £44.85. December. Acorn are still looking at what resident operational software to include.

**MIDI ADD-ON I/O PODULE.** Code: AKA15, £33.35. The User Guide is currently being printed, and it will be available in November.

**BACKPLANE.** Code: AKA01, £44.85. The Backplane includes the fan, and is currently in stock.

**WINCHESTER.** Code: AKD50, £573.85. December. Winchester upgrade includes backplane.

**2ND DISC DRIVE.** Code: AKD5, £143.75. October.

**MONO SYSTEMS.** Mono Monitor for use with the 305/310 systems not available until November. "Entry" systems will run equally well with any other standard mono monitor.

## BEEBUG

**"MODEM".** Hayes compatible modem podule especially for the Archimedes. Provides V21 & V23 baud rates as standard, with V22 & V22bis rates with tone dial. Full Hayes "AT" command set together with battery backed phone number and 'S' register store. Line monitoring through computer's speaker, V25 auto-answer and selectable bell tones. Price T.B.A. Available December.

**COMMAND.** Price T.B.A. (late 1987 - early 1988). Communications software using WIMPS, featuring a range of terminals - Viewdata, VT52, VT100, Tektronix 4010, Teletype etc. Supports range of modems, facility to customise modem drivers. Wide range of file transfer protocols including XMODEM, YMODEM, KERMIT, CET. Features disc manager, phone directory, host mode. Self-contained comms language to control all its features

**MONITOR CONVERSION KIT.** £40-£50. November. Enables the Archimedes to be connected to Microvitec and similar monitors. Will consist of a small interface box and leads.

## COMPUTER CONCEPTS

**INTERWORD,** £44.85. **INTERSHEET,** £44.85. **INTERCHART,** £28.75. ALL THREE £97.75. (£17.25 each to existing users). Emulator only. ROM only. Available November.

**WORDWISE PLUS.** An emulator version is available now at £33.35. Existing users may exchange their ROM for £11.50 by quoting their WW+ serial number. (Contact CC for details).

**INTERBASE.** Not to be released for the Archimedes. **DISC DOCTOR II.** According to Computer Concepts this product is not feasible on the current operating system. The anticipated availability is July 1988.

**"WORDPROCESSOR".** Available early 1988. Anticipated to be less than £100. Written specifically for the Archimedes. This will be on ROM. Similar but better than MacAuthor, with WIMPS, WYSIWYG, Magnify/Reduce, Auto-Indent, Multi-document, Multi-windows, Fonts, incorporation of graphics, spelling checker at 60,000 words per minute (an improved Spellmaster), Thesaurus, Multi Printer Drivers including Laser.

**"DRAWING PACKAGE".** Available on ROM in early 1988. Anticipated to be less than £100. Links to their wordprocessor and future products, will be fast, WIMPS, will include textual definitions of drawings which may be altered to change the drawing. It includes its own programming language.

**8 ROM PODULE.** Available November. This podule is

# News News News News News

intended to be compatible with Acorn's ROM board, but will be at a much lower price.

## CLARES

**ALPHA BASE.** £49.95. End of October. Compatible with the Model B's Betabase, free format input, window/icon driven, 400 fields, 27600 characters per record, range check, different access levels, passwords. Multiple pages/screens per record.

**IMAGE WRITER.** £29.95. Mid-October. A wordprocessor providing on-screen features reproducible on the Epson FX80, standard machine font, control window based, document (not page) based, graphics areas with graphics module.

## MINERVA SYSTEMS

**SYSTEM DELTA PLUS.** £69.96. Early November. Relational Database Management System. WIMP database allows search selection from within windows. Fully programmable using System Delta Commands. Programmer's Reference Guide available at extra £29.95.

**SUPER DELTA.** £199.95. First half of 1988.

**REPORTER.** £24.95. Early November. User definable reports, for use with Database and System Delta Plus.

**SYSTEM SIGMA.** Price T.B.A. November. Spreadsheet Management System.

**GAMMAPLOT.** £34.95. November. General graphics program allowing the production of graphics and charts such as bar, pie, scatter, line and histogram.

**PERSONAL ACCOUNTANT.** November. Requires either System Delta, or System Delta Plus to run. Personal Accountant consists of the following individual packages: Order Processing, Sales Ledger, Purchase Ledger, Nominal Ledger, Stock Management. £64.95 each.

**SPECIALIST APPLICATIONS.** Requires System Delta, or System Delta Plus to run. School Administrator £79.95, Newsagent £69.95. Ancestry, Timetabling Prices T.B.A.

## BBC SOFT

**ARCCOMM.** £24.95. January 88.

**ARCWORD.** Price T.B.A. No release date. This wordprocessor is to be aimed at the every-day, rather than the advanced user.

**BLACK QUEEN.** Approx £15. November. Black Queen is a Contract Bridge program available for BBC/Master & Compact. The Compact version will work on the Archimedes. A specialist Archimedes version will be released in December.

## WATFORD ELECTRONICS

**REAL TIME MONO DIGITISER.** Approx £170. November.

**SERIAL KIT.** £28.75. Available NOW. The kit enables the Archimedes to be linked with a BBC Model B or Master computer. 5.25"

*Note that Acorn has said that the use of a connecting cable to an external 5.25" disc drive will invalidate the guarantee, and may well damage your Archimedes.*

**DATABASE SOFTWARE.** Price T.B.A. Mid November.

**PAINT PACKAGE.** Price T.B.A. Late November. This package is Icon-driven and (according to Watford Electronics) is far superior to the Quest paint package.

## ACORNSOFT

**VIEW, VIEWSHEET, VIEWSTORE, VIEWINDEX.** Separate versions are available now that run under the emulator.

**VIEW PROFESSIONAL.** £113.85. Spring 1988. The full Archimedes version contains many extra features, and will possibly be re-named.

**LISP, £113.85. PROLOG, £113.85. LOGISTIX, £113.85.** All available now.

**TERMULATOR.** Available December.

## LOGOTRON LTD

**ARCHIMEDES LOGO.** £69. December 1987 Developed jointly with Acorn, this product is fully compatible with Logo for the Master and BBC computers. The INSTALL primitive allows Logo to be extended with specialised applications modules for example:

**CONTROL LOGO.** £17.25. Early 1988. Lets you explore computer control applications by talking to all the ports, as well as most interfaces and controllers.

**MUSIC LOGO.** £17.25. Early 1988. Takes full advantage of the Archimedes powerful sound potential, enabling you to use familiar musical notation.

**3D LOGO.** £17.25. Early 1988

**SOFT SPRITES.** Price T.B.A. Mid 1988

## AMS

**FINESSE PAINT.** Price T.B.A. Dec 1987. Finesse Paint is an art program based around a graphics tablet environment. It will develop into an integrated system including 3D CAD, Business Graphics, Maths Graphics and others.

# News News News

## SOLIDISK

**4 MEGABYTE BOARD.** Price T.B.A. The hardware is ready now, however, the complete product including software will not be available until the release of the new version of the operating system.

## UNILAB

**WEATHER SATELLITE RECEIVER & DISPLAY.** Price T.B.A. December 1987. Enables you to receive signals from the NOAA weather satellite and use the computer as a very sophisticated display terminal.

## TECHSOFT

**DESIGNER.** Approx £270. February 1988. CAD package for schools, colleges, small businesses. Priced similarly to the BBC/Master version which has been well received.

## ARMADILLO

**AUDIO SAMPLER PODULE.** Mid November 1987. Requires the Backplane. The board contains all the electronics to convert an audio signal into a digital one. An on-board EPROM contains the controlling software, the remainder is provided on floppy disc. Real-time display of voltage versus time; Spectral Analysis, Fast Fourier Transforms, Utterance detection, Dump to the VIDC sound chip.

## IAN COPESTAKE

**ARCHETYPE.** Price T.B.A. Early 1988 WYSIWYG wordprocessor capable of displaying and printing texts in a variety of linguistic and scientific character sets.

## MEADOW COMPUTERS

**MICRO-TRADER.** Approx £200. Fully integrated package including sales, purchase and nominal ledger. Integrates with Micro-Aid Extended Payroll. Pre-release version available early November, final version January 1988. Purchasers of the pre-release version will be able to upgrade in January.

**MICRO-STOCK.** Approx £75. Stock control package available as add-on or stand-alone.

**MICRO-MAILER.** Approx £30. Works with Wordwise or View.

RU

## WHAT IT DOES

The accompanying picture should give a reasonable idea of what the menu module offers. It will display all the contents of your current directory, with filenames in black and directory names in white, for easy identification. As you move the mouse over the various filenames the bottom line of the screen gives file details of the item currently under the pointer. You use two clicks of the *select* button (far left button) either to select a new directory, or to load and run a Basic program.

All file types other than Basic will be executed just as if you had entered:

\*filename

from the keyboard. Thus machine code will be loaded and run, modules will be installed, and so on. If on the other hand you double-click the *adjust* button (far right), the file will be simply loaded into memory.

Double clicking with any of the three mouse buttons on any of the upper boxes of the display will have the expected effect. *QUIT* will take you back to the environment which you were in when the menu was called (i.e. Basic, or Arthur, or whatever). *Dir ^* will take you up the directory hierarchy by one level, and *MOUNT* will mount a new disc. You can also use the latter to take you to the root directory of your current disc. Additionally, star commands can be entered directly from the keyboard.

## 305 USERS

This program will run on an otherwise default-configured 305 providing that the RMA size is configured to 10 or more.

## INSTALLING THE MODULE

Type the program in very carefully, and save it away to disc before running it. When you run the program, it will assemble the machine code module, then automatically save it away to disc under the filename *RMENU*, and

# ARCHIMEDES DISC MENU

By David Pilling



This piece of ARM code automatically installs itself as a relocatable module to give you mouse control of the Arc's disc environment. Move around the directories, load and run Basic programs, or load files into your word processor all at the click of the mouse.

then install the module. To call the menu, just type:

\*MENU

at any time. To leave the menu, either use the QUIT box, or press Escape.

The module will stay in place until you press Ctrl-Break, or the machine is turned off, and may even be called from within another application. To re-install it at any time, just type:

\*RMENU

If it is not in your current directory, you will need to give the full path name (for example \*\$.MODULES.RMENU).

I hope that you will find this menu so useful that you will want to automatically load it up from a !BOOT file every time that you switch your machine on. If so, watch this space, because next month we will be incorporating some extra facilities, like deleting marked files, etc. We will also put to use a special option for loading files into word processors and the like. This is already incorporated in the program, and is activated by double-clicking with the

middle button. And for those with tired fingers, there is a working copy of the menu program on this month's magazine disc.

```
10 REM >MenuP
20 REM Program      Disc Menu Module
30 REM Version      A 1.0P
40 REM Author       David Pilling
50 REM Risc User    November 1987
60 REM Program      Subject to Copyright
70:
80 DIM code &2000:PROCsetup
90 FOR pass= 0 TO 2 STEP 2
100 P%=code:[ OPT pass
110 EQUDO:EQUDO:EQUDO:EQUDO
120 EQUD help-code
130 EQUD help-code
140 EQUD comm-code
150 .help
160 EQU "Disc-Menu":EQUB0
170 .comm
180 EQU "MENU":EQUB0:ALIGN
190 EQUD go-code:EQUDO:EQUDO
200 EQUD helpt-code:EQUB0
210 .helpt
220 EQU "Use *MENU to engage the "
230 EQU "disc menu":EQUB13
240 EQU "Syntax: *MENU":EQUB0
250 EQU STRING$(128,CHR$0)
260.stack
270.poir
280 EQU "POINTER 1":EQUB13
290.go
300 STMFD R13!, {R0-R12,R14}
310 TEQP R15,#0:ADR R8,stack
320.go2
330 SWI ws:EQUB22:EQUB14:EQUB0
340 ADR R0,poir:SWI cli
350.newd:BL set
360 MOV R1,#4:MOV R9,#100
370.new1:BL moun:SUBS R1,R1,#1
380 BGE new1:BL bloo:SWI wr+31
390 SWI wr+3:SWI wr+7:SWI ws
400 EQU "Dir ":"EQUB0:ADR R2,blok
410 MOV R0,#6:BL plib:SWI wr+31
420 SWI wr+33:SWI wr+7:SWI ws
430 EQU "Lib ":"EQUB0:ADR R2,blok
440 MOV R0,#7:BL plib
450 BL getf:MOV R12,#77
460.mlp1:BL poin:LDRB R2,[R1,&14]!
470 CMP R2,#0:BNE mlp2
480 SUBS R12,R12,#1:BGE mlp1
490.mlp2:ADD R11,R12,#1
500 MOV R12,#0:B loopx
510.loop:BL tabn:BL poin
```

## ARCHIMEDES DISC MENU

```

520 LDRB R3,[R1,#10]:CMP R3,#2
530 BLEQ grog:BLNE blog:BL prif
540 ADD R12,R12,#1
550.lloopx:CMP R12,R11:BNE loop
560.mloop:SWI es:BCS quit
570 MOV R0,#1:STRB R0,rtcb
580 ADR R1,rtcb:MOV R0,#14:SWI wo
590 LDRB R0,rtcb+6:CMP R10,R0
600 MOV R10,R0:BLNE bloo:BLNE ptime
610 SWI ms:BL maps:BEQ mloop0
620 CMP R0,#&28:BLT mloop0
630 SUB R0,R0,#&28:CMP R0,R11
640 BGE mloop0:CMP R0,R9:MOV R9,R0
650 BEQ mloop0:MOV R12,R0:BL poin
660 BL bloo:SWI wr+31:SWI wr+0
670 SWI wr+24
680 SWI ws:EQUS" ":"EQUB0:BL prif
690 SWI ws:EQUS" Load:"EQUB0
700 LDR R0,[R1],#4:BL hex
710 SWI ws:EQUS" Exec:"EQUB0
720 LDR R0,[R1],#4:BL hex
730 SWI ws:EQUS" Length:"EQUB0
740 LDR R0,[R1],#4:BL hex
750.mloop0:MOV R0,#145:MOV R1,#0
760 SWI by:BCS mloop1
770 CMP R2,#ASC"~":"BNE mloop1
780 SWI ws:EQUB31:EQUB3:EQUB6
790 EQUS"~":"EQUB23:EQUB1:EQUB1:EQUB0
800 SWI wr:SWI wr:SWI wr:SWI wr:SWI wr
810 SWI wr:SWI wr:ADR R0,blok
820 MOV R1,#77:MOV R2,#32:MOV R3,#128
830 SWI rs:BCS ex:SWI wr+22:SWI wr+14
840 ADR R0,blok:SWI cli:SWI ws
850 EQUS"Press any key":EQUB0:SWI rd
860.ex:MOV R0,#&7C:SWI by:B go2
870.blok:EQUS STRING$(80,CHR$0)
880.mloop1
890 SWI ms:ANDS R2,R2,#7:BEQ mloop
900.mloop2:SWI ms:ANDS R2,R2,#7
910 BNE mloop2:MOV R0,#1:ADR R1,time
920 SWI wo:LDR R4,time
930.mloop3:MOV R0,#1:ADR R1,time
940 SWI wo:LDR R0,time:SUB R0,R0,R4
950 CMP R0,#35:BGTE msing:SWI ms
960 ANDS R2,R2,#7:BEQ mloop3:MOV R4,R2
970.mloop4
980 SWI ms:ANDS R2,R2,#7:BNE mloop4
990 SWI ms:BL maps:BEQ mloop
1000 CMP R0,#&28:BLT topco
1010 SUB R0,R0,#&28:CMP R0,R11
1020 BGE mloop:MOV R12,R0:BL poin
1030 LDRB R3,[R1,#10]:CMP R3,#2
1040 BEQ isdir:BL quits
1050 CMP R4,#1:BEQ load
1060 CMP R4,#2:BEQ menu
1070 LDR R0,[R1]:BIC R0,R0,#FF

1080 MOV R3,#0:SUB R3,R3,#500
1090 CMP R3,R0:BNE runc
1100 ADR R0,runbn:BL mbsb
1110 ADR R0,runb:SWI cli
1120.runc
1130 ADR R0,runn:BL mbsb
1140 ADR R0,run:SWI cli:B quit2
1150.run:EQUS"RUN "
1160.runn:EQUS STRING$(12,CHR$0)
1170.runb:EQUS"BASIC -CHAIN "
1180.runbn:EQUS STRING$(12,CHR$0)
1190.menu
1200 ADR R0,setn:BL mbsb
1210 ADR R0,setm:SWI cli
1220 ADR R0,exme:SWI cli:B quit2
1230.exme:EQUS"!MENU~":"EQUB13
1240.setm:EQUS"SET FNAME "
1250.setn:EQUS STRING$(12,CHR$0)
1260.load
1270 ADR R0,loadn:BL mbsb
1280 ADR R0,load:SWI cli:B quit2
1290.load:EQUS"LOAD "
1300.loadn:EQUS STRING$(12,CHR$0)
1310.isdir
1320 ADR R0,dirn:BL mbsb
1330 ADR R0,dir:SWI cli:B newd
1340.mbsb:ADD R1,R1,#14
1350.mbs1:LDRB R2,[R1],#1:cmp R2,#0
1360 MOVEQ R2,#13:STRB R2,[R0],#1
1370 BNE mbs1:MOV R15,R14
1380.dirb:EQUS"DIR ^":EQUB13
1390.dir:EQUS"DIR "
1400.dirn:EQUS STRING$(12,CHR$0)
1410.topco
1420 CMP R0,#&1D:BEQ quit
1430 CMP R0,#&19:BEQ back:SUBS R0,R0,#5
1440 BLT mloop:cmp R0,#5:BLT mountn
1450 REM more commands go here *****
1460 B mloop
1470plib:SWI gb:LDRB R3,[R2,#1]!
1480.lib1:LDRB R0,[R2,#1]!:SWI wc
1490 SUBS R3,R3,#1:BNE 1ib1:MOV R15,R14
1500.ptime
1510 SWI wr+31:SWI wr+53:SWI wr+0
1520 MOV R0,#0:STRB R0,rtcb:ADR R1,rtcb
1530 MOV R0,#14:SWI wo
1540 ADR R1,rtcb:MOV R2,#0
1550.ptlp:LDRB R0,[R1],#1:SWI wc
1560 ADD R2,R2,#1:cmp R2,#24:BNE ptlp
1570 MOV R15,R14
1580.rtcb:EQUS STRING$(25,CHR$0)
1590.mountn
1600 ADD R0,R0,#ASC"0":STRB R0,mouts
1610 ADR R0,mout:SWI cli:B newd
1620.mout:EQUS"MOU."
1630.mouts:EQUB32:EQUB13

```

```

1640.moun:ADD R0,R1,R1,ASL#2
1650 ADD R0,R0,R0,ASL#1:ADD R0,R0,#3
1660 SWI wr+31:SWI wc:SWI wr+1
1670 SWI ws:EQUS" MOUNT ":EQUB0
1680 ADD R0,R1,#ASC"0":SWI wc
1690 SWI ws:EQUS" ":"EQUB0:MOV R15,R14
1700.back
1710 ADR R0,dirb
1720 SWI cli:B newd
1730.msing
1740 REM insert single click code here
1750 B mloop
1760.quit
1770 MOV R0,#&7C:SWI by:BL quits
1780.quit2:SWI os
1790 LDMFD R13!,{R0-R12,R15}^
1800.quits:STMF D R8!,{R0-R7,R14}
1810 SWI ws:EQUB22:EQUB12:EQUB0
1820 MOV R0,#&6A:MOV R1,#0:MOV R2,#0
1830 SWI by:LDMFD R8!,{R0-R7,R15}
1840.time
1850 EQUD 0:EQUB 0
1860.maps
1870 RSB R1,R1,#1024:MOV R0,R0,ASR#4
1880 CMP R0,#3:BLT mapb
1890 CMP R0,#76:BGT mapb
1900 MOV R2,#0:SUB R0,R0,#2
1910.map1
1920 MOV R3,R0:SUBS R0,R0,#15
1930 ADDGE R2,R2,#1:BGE map1
1940 CMP R3,#0:BEQ mapb:MOV R0,#0
1950.map2:SUBS R1,R1,#40:ADDGT R0,R0,#1
1960 BGT map2:ADD R0,R0,R0,ASL#2
1970 ADD R0,R0,R2:ORRS R1,R1,#1
1980 MOV R15,R14
1990.mapb
2000 SUBS R1,R1,R1:MOV R15,R14
2010.hex
2020 STMF D R8!,{R0-R7,R14}
2030 MOV R3,#8:MOV R1,R0
2040.hex0
2050 MOV R1,R1,ROR#28
2060 AND R0,R1,#15:CMP R0,#10
2070 ADDGE R0,R0,#ASC"A"-10
2080 ADDLT R0,R0,#ASC"0":SWI wc
2090 SUBS R3,R3,#1:BNE hex0
2100 LDMFD R8!,{R0-R7,R15}
2110.tabn
2120 SWI wr+31:MOV R0,R12:MOV R2,#0
2130.tabl:ADD R2,R2,#1:MOV R1,R0
2140 SUBS R0,R0,#5:BGE tabl:MOV R3,#15
2150 MUL R0,R1,R3:ADD R0,R0,#3:SWI wc
2160 ADD R0,R2,#7:SWI wc:MOV R15,R14
2170.prif
2180 ADD R3,R1,#14
2190.prif2:LDRB R2,[R3],#1:CMP R2,#0
2200 MOV R0,R2:SWINE wc:BNE prif2
2210 SUB R3,R3,R1:RSB R3,R3,#&21
2220.prif3:SUBS R3,R3,#1:SWIGT wr+32
2230 BGT prif3:LDRB R3,[R1,#&C]
2240 LDRB R2,[R1,#10]
2250 CMP R2,#2:BNE nodd
2260 SWI ws:EQUS" D":EQUB0:BEQ noss
2270.nodd
2280 TST R3,#1:SWINE wr+ASC"R"
2290 SWIEQ wr+32:TST R3,#2
2300 SWINE wr+ASC"W":SWIEQ wr+32
2310.noss
2320 TST R3,#8:SWINE wr+ASC"L"
2330 SWIEQ wr+32:MOV R15,R14
2340.poin:ADR R1,buff:MOV R0,R12
2350.poil:SUBS R0,R0,#1:MOVMI R15,R14
2360 ADD R1,R1,#&14
2370.poi12:LDRB R2,[R1],#1:CMP R2,#0
2380 BNE poi12:SUB R1,R1,#1
2390 BIC R1,R1,#3:ADD R1,R1,#4:B poil
2400.getf
2410 ADR R0,buff:MOV R1,#&F00:MOV R2,#0
2420.ml1p0:STR R2,[R0,R1]:SUBS R1,R1,#4
2430 BPL ml1p0:MOV R0,#10:ADR R1,null
2440 ADR R2,buff:MOV R3,#77:MOV R4,#0
2450 MOV R5,#1000:MOV R6,#0:SWI gb
2460 MOV R15,R14
2470.null:EQUB0
2480.bloo
2490 SWI ws:EQUB17:EQUB128+10:EQUB17
2500 EQUB0:SWI wr+0:MOV R15,R14
2510.blog
2520 SWI wr+17:SWI wr+0:SWI wr+17
2530 SWI wr+128+8:MOV R15,R14
2540.grog
2550 SWI wr+17:SWI wr+11:SWI wr+17
2560 SWI wr+128+8:MOV R15,R14
2570.set:ADR R1,sets
2580.set1
2590 LDRB R0,[R1],#1:CMP R0,#ASC"$"
2600 SWINE wc:BNE set1:MOV R15,R14
2610.sets
2620 EQUB19:EQUB0:EQUB24:EQUB64:EQUB80
2630 EQUB112
2640 EQUB19:EQUB8:EQUB16:EQUB128
2650 EQUB144:EQUB176
2660 EQUB19:EQUB9:EQUB16:EQUB0:EQUB48
2670 EQUB180
2680 EQUB19:EQUB10:EQUB16:EQUB0:EQUB240
2690 EQUB240
2700 EQUB19:EQUB11:EQUB16:EQUB224
2710 EQUB224:EQUB240:EQUB17
2720 EQUB128+9:EQUB12:EQUB17:EQUB128+10
2730 EQUB17:EQUB0:EQUB31:EQUB0:EQUB24

```

continued on page 26



# ARCHIE'S WHITE MOUSE

In the first of our series on graphics, Lee Calcraft shows how to make the Archimedes mouse perform to order.

A305 SC - SP - RMA 1D or more

The Archimedes mouse is a neatly constructed little rodent which software packages like the desktop, the painting program and the music editor on the Welcome disc have exploited to the full. Fortunately, Acorn have provided a simple set of commands which may be used from Basic to manipulate the mouse in any way that the user wishes. However, as with most other aspects of the A series machines, documentation is minimal, and the purpose of this article is to take you through some of the commands, and to develop a simple mouse-driven application.

## POINTER ON

In most cases when the mouse is in use, its position is indicated by an on-screen pointer, and this must be specifically enabled in order to make it visible. To do this, type:

\*POINTER

You should now see the familiar blue pointer appear on screen; and moving the mouse around should cause the pointer to move. Once the pointer has been enabled, the mouse and pointer system can be turned on and off with the two commands:

MOUSE ON

and      MOUSE OFF

You will also see that changing screen mode clears the pointer (reinstate it with MOUSE ON), but that executing:

CLS

or      CLG

to clear the current screen leaves the pointer unaffected.

## COLOURING THE POINTER

Although the standard pointer comes in two-tone blue, a special command is provided to alter its colour. This has the syntax:

MOUSE COLOUR n,16\*r,16\*g,16\*b

where n is the so-called *logical colour* of the part to be re-coloured. In practice n=1 refers to the outline of the default pointer, and n=2 the inside colour. The three remaining parameters

## MOUSE COMMANDS COVERED

\*POINTER  
MOUSE ON  
MOUSE OFF  
MOUSE COLOUR  
MOUSE RECTANGLE  
MOUSE TO  
MOUSE

r, g, and b are the red, green and blue components of the new colour; and each must be supplied with a sixteen times multiplier, as implied in the definition above. The amount of each component r, g, and b can range from 0 to 15 in integer steps. Thus:

MOUSE COLOUR 1,0,240,0

will set the outline of the pointer to bright green (i.e. full green, with no red or blue), the 240 being derived from  $16*15$ . If you try:

MOUSE COLOUR 2,176,176,176

this will set the inside of the pointer to a delicate shade of grey. If you do not like the colour combination that you have set at any time, you can just execute:

\*POINTER

to reset the colours to Acorn's choice.

## RESTRAINING THE POINTER

As you can see from experiment, the pointer can be made to travel anywhere within the main screen area. But you can if you wish restrict it to any chosen rectangle within that area. There are many reasons why you might wish to do this. For example, in a drawing package, you may want to restrict the pointer to the main drawing area, and so on. Again there is a simple command from Basic to achieve this. Contrary to information given in some versions of the User Guide, it takes the form:

MOUSE RECTANGLE x,y,width,height  
where x and y are the co-ordinates of the bottom left hand corner of the rectangle, measured in graphics units; and width and height are simply the width and height of the rectangle, again in graphics units.

The graphics units referred to are those used in all graphics modes of the Archimedes, and go from 0 to 1279 along the x (or horizontal) axis, and from 0 to 1023 along the y (or vertical) axis. The origin, or 0,0 point, is at the bottom left of the screen. To test this out, try the following:

MOUSE REC. 400,400,300,300

You should find that the pointer is now confined to a small rectangle at the centre of the screen. Notice also the use of the abbreviation of the keyword.

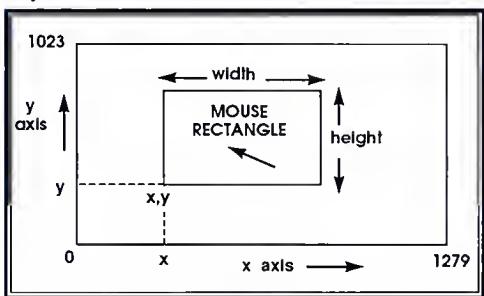


Figure 1.

#### Definition of MOUSE RECTANGLE.

#### DIRECTING THE POINTER

In a similar way you can send the pointer to any part of the screen - providing that it is not outside the area defined with MOUSE RECTANGLE. The command:

MOUSE TO x,y

will move the pointer to the position x,y, where x and y are again given in graphics units, and can range from 0 to 1279, and 0 to 1023 respectively. Try:

MOUSE TO 0,0

This should cause the pointer to move to the origin at the bottom left of the screen.

#### RETURNING DATA

So far we have only looked at how to do things *to* the pointer: to colour it, move it around, and so on. The most useful bit is still to come. There is a simple command to get data back *from* the pointer. This informs us not only of where on the screen the pointer currently resides, but also what buttons, if any, are being

pressed. Clearly this is all vital information, and it is fortunately very easily gathered. The command:

MOUSE x,y,b

does it all. Each time that this function is executed, it returns the values of the x co-ordinate, the y co-ordinate, and the button pressed status in the variables x, y and b respectively. To see how this works, try the simple program which appears in listing 1.

#### Listing 1

```

10 REM >mousetest2
20 REM Mouse test
30 MODE 8
40 OFF
50 VDU19,0,24,0,240,240
60 CIRCLE 640,512,10
70 *POINTER
80 REPEAT
90 MOUSE x,y,b
100 PRINT TAB(30,5) "X=";x;SPC4
110 PRINT TAB(30,6) "Y=";y;SPC4
120 PRINT TAB(30,7) "b=";b;SPC4
130 UNTIL FALSE

```

When you run this program you should see a cyan-bordered screen with a small circle at the centre, and three numbers printed on the screen. These are the two co-ordinates of the mouse, its grid reference if you like, plus the result returned by the button status. You should find that the top two numbers, the x and y co-ordinates of the pointer position, change as you



move the mouse around. When you move it to the left, the top number, the x co-ordinate will decrease, and as you move it to the right, it will increase towards the maximum value of 1279. The second figure, the y co-ordinate, decreases as the pointer is moved downwards, and increases to a maximum of 1023 as it is moved upwards. The small circle

## ARCHIE'S WHITE MOUSE

in the centre of the screen should correspond to co-ordinates 640, 512.

Now see what happens if you press a button. The button value, formerly zero, increases. If you press the left hand button, the *select* button, the number 4 should appear, the middle or *menu* button produces a 2, while the right hand button, the *adjust* button, produces a 1. You will also see that if you press different buttons in combination, the numerical result will be the sum of the individual buttons pressed. To exit the program, just press Escape.

As you can see from the listing, the program is reasonably straightforward. Line 30 sets the screen to mode 8 (a high resolution graphics mode which is reasonably economical on memory), then line 40 switches off the flashing cursor. Line 50 sets up the border colour, to delimit the screen area. For more details on this command, see this month's hints. Line 60 draws a small circle at the centre of the screen, and the following line initialises the pointer. We now come to the main loop of the program. This is a REPEAT-UNTIL loop which continuously reads the mouse data (line 90), and displays the results.

### BUTTON VALUES RETURNED

- - -	0	- M A	3
- - A	1	S - A	5
- M -	2	S M -	6
S - -	4	S M A	7

S=Select (Left button)  
M=Menu (Middle button)  
A=Adjust (Right button)

Figure 1

### MOUSE SKETCH PAD

Armed with this basic information we can write a simple drawing program based around the mouse. The completed program is given in listing 2. It is relatively short, but performs well enough for simple sketching and doodling. If you type it in and run it, you will see a cyan border, with the painting area taking up the rest

of the screen. The controls are quite simple, as may be seen from figure 2. With a little practice you will find that they are quite easy to use.

### SKETCH PAD CONTROLS

S - -	Fine Line
- M -	Medium Line
- - A	Thick Line
S M -	Eraser
- M A	Clear Screen
S - A	Load or Save Screen
S M A	Exit

S=Select (Left button)  
M=Menu (Middle button)  
A=Adjust (Right button)

Figure 2

The only option which is not self explanatory is the load/save option invoked by pressing the select and adjust buttons simultaneously. This sets the screen border to red, and then waits for a key from the keyboard to be pressed. If S is pressed, the current screen will be saved away. If L is pressed, a screen is loaded from disc, while if any other key is pressed, the border is returned to cyan, and the sketching mode resumed. The filename used for both save and load operations is always *screen*, and you should be warned that even though we have used the proprietary commands, saving and loading take a very long time. Also, since no special error trapping has been included in the program, if a disc error occurs, the program will end, though without clearing the screen. Finally, the program may be exited by pressing all three mouse buttons simultaneously. To do this, you should start by pressing the middle button, to avoid calling the load/save option by mistake.

### HOW IT WORKS

Let us take a look at how the program works. Lines 80 to 100 will be familiar from listing 1. But inside the REPEAT loop at line 120 there is now a so-called CASE statement. This is situated between lines 150 and 220, where the keyword ENDCASE indicates the

exit point. The case statement is used to perform a number of different actions depending on the state of the variable b. That is the effect of line 150:

CASE b OF

Line 160 handles what happens if b=4, line 170 if b=2, and so on.

When b=4 this means that the select button has been pressed. In response to this we draw a point at the pointer's own co-ordinates. This is performed by:

POINT x,y

This will draw a fine line if the button is held down, and the mouse moved. To draw a thicker line, we use a filled circle with a very small radius. This is accomplished in line 170 in response to b=2 - the pressing of the middle mouse button. A circle with a larger radius is used in line 180 to draw the thick lines.



Line 190 implements the eraser by switching the foreground drawing colour to black, with GCOL 0, and drawing thick lines. Line 200, as you can see, implements a screen clear when the two rightmost buttons are pressed. The procedure PROCdelay, defined at line 270, is used by both the eraser and the clear screen options. Its purpose is to ensure that no extra blobs get drawn after clearing or erasing, once the mouse buttons are released. Finally, line 210 implements the load and save option, calling the procedure PROCladsave at line 320.

As you can see, the main part of the program between lines 80 and 250 is extremely short. This illustrates, I hope, just how easy it is to create quite effective programs based around the mouse. You may care to write something along similar lines, or to extend the Sketch Pad given here into a full-feature paint program, where painting and colour functions are selected by moving to specific boxes placed at the border of the screen. This of course opens up an infinity of options, rather than being limited to the number of different combinations of the three mouse buttons.

#### Listing 2

```

10 REM >Sketch5
20 REM Program    Mouse Sketch Pad
30 REM Version   A 0.5
40 REM Author    Lee Calcraft
50 REM Risc User November 1987
60 REM Program   Subject to Copyright
70 :
80 MODE 8:OFF
90 VDU19,0,24,0,240,240
100 *POINTER
110 a=0
120 REPEAT
130 MOUSE x,y,b
140 IF b<>6 AND a=6 THEN PROCdelay
150 CASE b OF
160 WHEN 4:POINT x,y
170 WHEN 2:CIRCLE FILL x,y,5
180 WHEN 1:CIRCLE FILL x,y,20
190 WHEN 6:GCOL 0:CIRCLE FILL x,y,20:G
COL 7
200 WHEN 3:CLS:PROCdelay
210 WHEN 5:PROCladsave
220 ENDCASE
230 a=b
240 UNTIL b=7
250 ON:END
260 :
270 DEFPROCdelay
280 TIME=0:REPEAT UNTIL TIME>50
290 MOUSE x,y,b
300 ENDPROC
310 :
320 DEFPROCloadsav
330 VDU19,0,24,240,0,0:*FX15
340 A=GET:CASE A OF
350 WHEN 83:OSCLI("SCREENSAVE screen")
360 WHEN 76:OSCLI("SCREENLOAD screen").
370 ENDCASE:VDU19,0,24,0,240,240
380 ENDPROC

```

RU

# Painting

FOR THE ARCHIMEDES =



With the Archimedes' superb graphics capability it is hardly surprising that Acorn should include a painting program on the Welcome disc, and Clares has quickly followed suit with its Artisan. How good are these two packages, and is it worth spending nearly £40 on the latter when the former is supplied free with the Archimedes? Mike Williams helps you make up your mind.

One might have expected that Acorn would have dreamt up a more imaginative name for the painting package supplied on the Welcome disc. After all, the speed and colours of Archimedes do have enormous potential for graphics. Notwithstanding, the Paint package does provide a most workmanlike approach, offering many of the facilities that one might expect. Don't be put off either by the unexciting screen layout; the facilities are there and they are easy to use.

## Welcome Paint

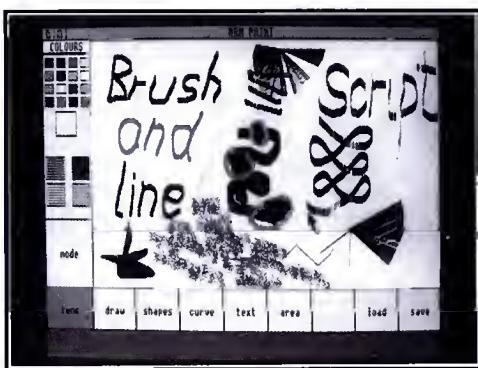
If you have an Archimedes no doubt you have taken a look at what Paint has to offer. However, there are some less obvious features you may have missed, and we'll review the main features for those who do not yet have an Archimedes, or haven't tried this program.

Paint makes little use of windows and icons. Instead, various options are displayed to the left and right of the main drawing area, and are selected with the mouse. To the left of the screen the current set of colours is displayed, 16 in all, together with four patterns. The mouse may be used to select any colour or pattern as the current drawing colour. What is not readily apparent is, that by pointing at the word Colour at the head of this colour menu, will superimpose on the drawing area a colour mixer allowing any one of the Archimedes available colours to be included among the 16 on the screen. However, changing any colour in this way will also change any example of this colour already on the drawing area.

The controls for the colour mixer are a circle and a bar. Moving a radius around the circle controls the basic choice of colour with further variations achieved by changing the length of the radius. Moving the bar slider up and down appears to darken or lighten the selected hue. The approach is a bit weird, and very much inferior to the colour palette control that is part of the desktop. This method of colour mixing is also incorporated in Artisan, and looks like setting the standard for this facility on the Archimedes.

There is also a mode box which performs two functions. From a sub-display, you can select solid or outline shapes (called fill or open), and the mode of plotting (XOR, OR, AND or INV). As with other sub-menus, clicking on the diagonal cross symbol at the top of the window is the only way of removing it from the screen, unlike Artisan where menus automatically disappear once the mouse pointer moves away from them.

The bottom edge of the screen displays the main functions of Paint. Selecting any of these will similarly display an appropriate sub-menu along the bottom of the screen. When you have finished using a particular function, clicking on the function name in the bottom left-hand corner will return you to the main function menu, again not an obvious step.

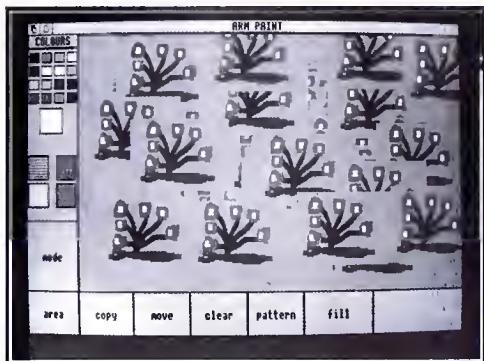


## DRAWING AND PAINTING

The draw function is likely to be the most used, and this provides 5 variations (line, joined, radii, brush, select). Line and joined both provide rubber-band style drawing, the former as separate lines (click at start and end), and the latter joined lines as the name implies. The last two options are concerned more with brush painting, allowing various shapes and sizes of brush to be selected. The sizing is not very extensive, and surprisingly no round brushes are provided, only triangular, rectangular and similar. In addition, I found the



brush and select options, and the select sub-menu somewhat cumbersome in use.



## SHAPES AND CURVES

The shapes and curve functions can clearly be grouped together. These provide a range of straight-line and curved shapes, either outline or filled depending on the current mode selected. All the shapes, as you might expect on an Archimedes, are drawn very swiftly indeed. There is also a text option which allows text, but in one height and style only, to be positioned on the screen.

## AREA

The area function is one of the most useful, and some of the options within this function can produce quite fascinating results. There is a very fast flood fill which works from the current pointer position. A clear screen function may also be selected. This appears to do nothing till you move to the main drawing area and click the mouse - the screen immediately clears with no further warning!

The Copy and Move options are much more interesting. In both cases a rectangle is defined to enclose part of the screen drawing area. This can then be moved or copied to any other location on the screen. The Copy function in particular, especially if the mouse is moved around with the button pressed, can produce most interesting displays very quickly indeed. This is a feature that is well worth exploring, but similar and much more extensive facilities are provided by Artisan. Lastly there is a function to allow you to define your own patterns on a pixel by pixel basis.

## LOAD AND SAVE

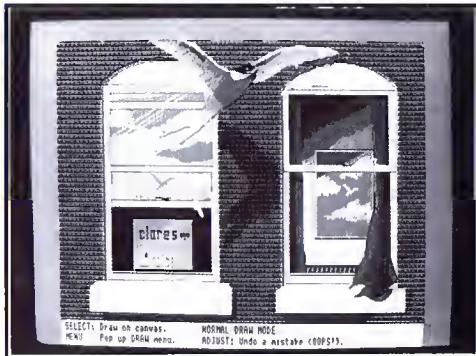
These provide the facility to save and load screen displays. There is no apparent way of using star commands from within the Paint package, so saving and loading must be on the same disc as the Paint package. The omission of this facility by Acorn does seem a bit remiss.

## Artisan

Clares, £39.95 inc VAT

Artisan is a new painting and drawing program for the Archimedes from Clares, and perhaps derives something of its ancestry from Artist, produced by Clares for Acorn's Compact, and bundled in the Italian version of the same machine sold by Olivetti in Italy.

Artisan is a highly superior product providing full colour graphics capability through some extremely well designed and colourful menus. These are very classy, with a three-dimensional look to them, and superb examples of what can be achieved with an Archimedes. We have not yet seen the final version of this product, but the pre-release version certainly impresses.



Four main menus are implemented at present. One of these controls such aspects as choice of brush, colour selection, flood fill, colour mixing and pattern defining. This menu also provides a three level zoom facility (two, four or eight times magnification), allowing individual pixels to be changed. Colour mixing and pattern definition, for example, lead into further sub-menus. As a comparison with Welcome Paint, Artisan provides a rounded brush whose size may be varied



continuously throughout the available range of sizes. Furthermore, there is a facility to define your own brush shape leading to all manner of effects (striped paint for example).

The mouse buttons are used to the full, with the left-hand button selecting from menu options, while the middle button calls up the current menu on the screen, and the right-hand button is used in specific functions, such as zooming in and out.



A second menu provides a variety of filled and outline shapes, and the opportunity to define your own shape. As with the others, the user-defined shape can be positioned and manipulated on screen to achieve the desired size and orientation. Separate and linked lines using rubber banding are readily drawn, and both this menu and the previous one include their own colour selection functions.

A further menu screen provides for the saving and loading of screens, and Clares has provided four superb examples to inspire and motivate. It is also possible to save and load patterns and sprites, although the latter feature was not implemented in our early review copy. The final menu is in many ways the most interesting, and provides some fascinating variations on the block copy and move idea. Any part of a picture can be copied or moved, and enlarged, reduced, rotated or flipped (to give a mirror image) at the same time. Several special effects are included, such as the magic pen, which allows one or more specified colours only to be changed by moving the magic pen across the screen. Such features are difficult to describe in a few words, but following the instructions in the

accompanying manual soon begins to bring home the enormous potential of Clares' Artisan.

## CONCLUSIONS

The Welcome Paint package provides most of the functions one might expect of a basic drawing package, and performs a workmanlike if uninspiring job. More imaginative use of windows and icons, of which Archimedes is fully capable, would have done much to lift Paint above the merely ordinary. A zoom feature to allow pictures to be manipulated at the pixel level is really essential for any serious use, while the opportunity to display text in different styles, sizes and angles would have helped as well. Lastly, one or two examples of what might be achieved with the Paint package would have done much to motivate the budding artist. However, Paint doesn't cost you a penny (once you have bought an Archimedes), so if you just want to play about it will probably do fine.

Although both Paint and Artisan have many features in common, including the use of mode 12 and its limitation of 16 colours, Artisan is much slicker and a lot more fun to use. The automatic deletion of menus in Artisan as soon as the pointer moves away does take some getting used to, but will be seen as an advantage as one's skill develops. I was very impressed with almost everything I tried with Artisan, and the manual (provisional version only so far) clearly explained all the many features with examples based on the five pictures included on the disc. If computer art is to be fun then go and buy this package now - you won't be disappointed. We can only wait in eager anticipation for the final version.

RU

Clares Micro Supplies, 98 Middlewich Road, Rudeheath, Northwich, Cheshire CW9 7DA.

## RISC USER PAINTING COMPETITION

Send us on disc your best artistic efforts, achieved with the Welcome Paint program, and we will award a prize of Clares' Artisan to the one we judge to be the best. Send your entries to:

**RISC User Painting Competition,**  
**Dolphin Place, Holywell Hill, St Albans,**  
**Herts AL1 1EX.**

All discs will be returned where postage is included.  
Closing date 5 December 1987.

# SOUND AND MUSIC

By David Graham

A brief foray into the Archimedes Sound commands,  
including details on how to load new instruments.

## BASIC SOUND

The fastest way to get sounds out of the Archimedes is to use VDU7 or to press the Ctrl key at the same time as the "G" key. This produces a single standard beep. To get anything more you need to use the SOUND command. In its simple form, this works just like the earlier BBC B and Master computers. Four parameters are used to give the sound channel, the volume, the pitch and duration respectively. For example, try:

SOUND 1,-15,200,20

or for something lower in pitch:

SOUND 1,-15,100,20

or quieter:

SOUND 1,-8,100,20

This may look a bit involved if you are not used to the BBC micro, but it is fairly straightforward.

The Sound Command	
Syntax:	
SOUND,Channel,Volume,Note,Duration	
Parameter	Range
Channel	1 to 8
Volume	-15 to 0
Pitch	0 to 255
Duration	0 to 255 (in twentieths)

Figure 1

Let us take a closer look at the four parameters in turn. The first, the channel number, can take values from 1 to 8, but only channel 1 is normally present at switch-on. This contrasts with the model B which has only 4 channels, but all are active at power-up. The volume parameter takes the range -15 to 0, with zero producing no sound, and -15, maximum. The pitch parameter should be in the range 0 to 255, with high values producing highest pitch. The User Guide (page 150 in the

June 1987 edition) gives the correspondence between the pitch parameter and the actual note played.

Finally the fourth parameter determines the duration of the sound produced, and is given in units of twentieths of a second. In our example we used a value of 20, and so produced a note of one second's duration. Integer values between 0 and 255 may be used. When 255 is used it is said to turn the sound channel on indefinitely (i.e. until Escape is pressed); though in practice the effective duration of the sound will not exceed the natural decay of the particular voice which is in use. It may now be worth experimenting with the volume, pitch and duration parameters for a bit before proceeding further.

## LOADING NEW VOICES

As I said, the Archimedes has only one sound channel active on power-up. But it is possible to configure up to eight channels for simultaneous use. Each sound channel can be allocated to one so-called *voice*. To see what voices are available on your machine, and how they are allocated, just type:

\*VOICES

This will cause a *Channel Allocation Map* to be displayed. It will probably look like that shown in figure 2. This shows that only a single voice is present called *WaveSynth-Beep*, and that it has been allocated to channel 1. That is why we have restricted ourselves to channel 1 so far.

Channel Allocation Map before loading voices	
Voice	Name
1	1 WaveSynth-Beep
~~~~~ Channel Allocation Map	

Figure 2

# SOUND AND MUSIC

Fortunately the Welcome disc contains some new voices. To load these in, first quit Basic using:

QUIT

Then, at the asterisk prompt, type:

```
RMLOAD $.MODULES.stringlib  
and
```

```
RMLOAD $.MODULES.percussion  
and get back into Basic with:
```

BASIC

If you now check the allocation map using:

\*VOICES

you should see a display similar to that in figure 3. There are now 9 resident voices, though just as before, only voice 1 is allocated (again to channel 1).

Channel Allocation Map after loading voices	
Voice	Name
1	WaveSynth-Beep
2	StringLib-Soft
3	StringLib-Pluck
4	StringLib-Steel
5	StringLib-Hard
6	Percussion-Soft
7	Percussion-Medium
8	Percussion-Snare
9	Percussion-Noise
~~~~~ Channel Allocation Map	

Figure 3

To allocate some of the others, proceed as follows:

\*CHANNELVOICE 2 4

\*CHANNELVOICE 3 8

This sets channel 2 to the "Steel" string voice (voice 4), and channel 3 to the "Snare" drum (voice 8). Type \*VOICES to see the new allocation.

Before we can make use of these new voices, we must tell the system how many voices we want to make active. To enable the

new voices, type:

VOICES 3

This last operation is easily forgotten. It is doubly confusing because it has the same name as that used to display the allocation map. But this time it is not a star command it is part of ARM Basic - i.e. we must use VOICES rather than \*VOICES. In fact the sound system allocates voices in pairs, and VOICES 3 has the same effect as executing VOICES 4.

## Loading in New Voices

```
QUIT      (temporarily leave Basic)  
RMLOAD $.MODULES.stringlib  
RMLOAD $.MODULES.percussion  
BASIC    (get back into Basic)  
*CHANNELVOICE 2 4 (allocate voices)  
*CHANNELVOICE 3 8  ditto  
*CHANNELVOICE ... (allocate more  
..... as required)  
VOICES 3 (up to 8, as applicable)
```

Figure 4

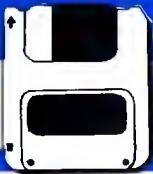
Now we are ready to test out some of the new sounds. Try a few variations:

SOUND 2,-15,100,30

SOUND 3,-15,1,10

You can also allocate more channels to some of the other voices, and test out the result. You will find, somewhat disappointingly, that the pitch parameter has no effect on any of the percussion sounds (though at least duration does), so you might as well set the pitch to 1 each time. You may also find the repertoire of voices a little disappointing. In fact all the voices supplied were generated mathematically, rather than from sampling real instruments. Fortunately though, at least one supplier is producing a sound sampler for the Archimedes. This will be able to create sound libraries directly from real sounds, and should result in much more realistic and interesting effects.

RU



# USING DISCS

By Lee Calcraft

If you are a newcomer to using Acorn's ADFS disc system, then this introductory article should help to get you started.

The Archimedes 305 and 310 computers both incorporate a single 3.5 inch floppy disc drive. All operations using this drive are handled by a reasonably sophisticated disc system called the Advanced Disc Filing System, or ADFS for short. This article is about using the ADFS.

If you have not used a disc system before, then there is a good deal to get to grips with, and you are recommended to take a look at section 9 of the User Guide, which provides useful introductory information. In particular, you will need to be familiar with the idea of disc directories, and the hierarchical way in which they are organised. But let us start with disc files themselves. All data stored on disc, whether it be a Basic program, a block of text or a screen dump, is held as individual files. These files are themselves organised into groups called directories, each of which can accommodate up to 47 or 77 files depending on the disc format in use.

## FORMATTING

Before a disc can be used for the first time it must be formatted. This process lays down magnetic tracks on which programs and other data may later be stored. As the User Guide explains, the Archimedes can provide two alternative formats; one capable of storing up to 640 kilo bytes of data (=655360 bytes) and allowing up to 47 objects per directory, or a so-called double density mode capable of storing up to 800 Kilo bytes of data (=819200 bytes) with up to 77 objects per directory. In fact the latter option has both greater capacity, and greater speed than the former, which is merely retained for compatibility with the ADFS on the old BBC micro (including Master and Compact). The formatting commands are:

\*FORMAT 0 L

or \*FORMAT 0 D

The zero specifies the disc drive number, and the following letter supplies the format size

required. D indicates the double or extra large mode. When you have issued this command, you will be asked to confirm your intention. This is because formatting a disc destroys everything which it previously held.

Whichever disc format you are using on the Archimedes, it is essential to employ high quality double sided discs. These should have been approved for use at 135 TPI (or tracks per inch). In all cases, it is possible to use discs of a lower rating or quality, but by doing so you will considerably increase the chances of disc failure, and the consequent loss of programs and data.

## SAVING FILES

After formatting a disc, the Archimedes will automatically verify it to check that it has formatted correctly, and will report accordingly. If all is well, you are now ready to save some files. For the purposes of experiment you could create a one line program as follows:

10 PRINT "A dummy program"

To save it, type:

SAVE "name"

where *name* is any name of up to 10 characters in length. For the sake of argument we will call it "Prog1".

If you now type:

\*CAT

this will display a catalogue of the disc, which should look something like that in figure 1. The program file Prog1 is shown to be present, and you will see that its name is followed by two letters WR. These indicate that the file is enabled for both read and write operations - in other words it is not locked. You should now be able to run the program directly from disc by typing:

CHAIN "Prog1"

Returning to the catalogue display, take a look at the \$ sign at the left of the screen. This is the name of the directory currently in use.



## USING DISCS

```
$ Disc 11_24 Mon :0
Option 00 (Off) URD "Unset"
Dir. "Unset" Lib. "Unset"
Prog1 WR
```

Figure 1. Catalogue display

### DIRECTORIES

When you format a disc, it is provided with just one directory, the so-called *root* directory. This is indicated by the \$ sign. It is then up to the user to create further directories as he wishes. One of the main advantages of doing this is that it allows you to organise your various programs and data in a sensible manner. For example, you might wish to use one directory for all correspondence, one for graphics programs, one for utilities, one for business applications, and so on. If you generate a lot of correspondence, you may wish to use a separate correspondence directory for each month of the year. The system is completely flexible, and it is entirely up to you how you organise things.

So how do you create a new directory? Just use the command:

\*CDIR name

where *name* is the name of the new directory. As with filenames, directories may be of up to 10 characters in length. As an experiment, try:

\*CDIR TESTDIR

If you now catalogue your disc with:

\*CAT (or \*. for short)

you should see something like that shown in figure 2. The new directory appears alongside the file that we created earlier, but it differs from it in having the letters DL after it. The D indicates that it is a directory, and the L indicates that it has been locked and so may not be deleted or over-written.

```
$ Disc 11_24 Mon :0
Option 00 (Off) URD "Unset"
Dir. "Unset" Lib. "Unset"
Prog1 WR TESTDIR DL
```

Figure 2. Catalogue display

If we wish to select this new directory we can use the command:

\*DIR TESTDIR

This makes TESTDIR the *currently selected* directory. If you now perform a

\*CAT

you can see the contents of this new directory, which of course should be empty. If you save a program now, it will be saved into the TESTDIR directory because that is the currently selected directory. You can get back to the root directory by simply typing:

\*DIR \$

Once you are back in the root directory, and you wish to save or load a program in the TESTDIR directory, you can either do it by re-selecting that directory (using \*DIR TESTDIR) or by loading (or saving) directly, specifying the full path to that directory, thus:

LOAD "\$.TESTDIR.progname"

Note the use of the full stops to separate the directory names.

As you may have realised, the new directory TESTDIR which we created, actually resides within the old root directory; and you may begin to see the reason for calling it a root directory. All other items stem from the root directory. If you create new directories, they will be within the root directory, or indeed within other directories, which are themselves within the root directory. This is called a *hierarchical* directory structure.

Figure 3 shows the contents of an ADFS disc, arranged so that you can see the so-called tree structure more clearly. From this you can see that the root directory contains just two objects, a file called Prog1 and a directory called MAINPROGS. The directory MAINPROGS itself contains two sub-directories (UTILITIES and APPS), and three files. For clarity, directories have been created using upper case text, and you may like to follow this convention. Directories can in fact be nested down to 127 levels, and each directory, except the last, could contain 77 further



directories, so you can see that an ADFS disc can have a very complex structure. In practice it is a good idea not to nest your directories too deeply, because it makes moving around within the directory structure quite laborious.

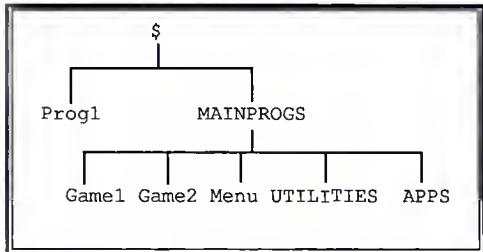


Figure 3. ADFS directory hierarchy.

## SHORT CUTS

There are also other ways to make life a little easier. The first is to use the disc menu listed elsewhere in this issue. Once this is installed in your machine, you need only type:

\*MENU

and then use the mouse to move around within the directory structure of your disc. You can run program files directly from the menu, as described in the article, or if you exit from the menu, you will find that you are left in the directory which you last selected.

Another trick is to use the function keys to take you to your most commonly used directories. See figure 4 to give an idea of how this is done. (The bar character | is on the key immediately above the Return key.) You will see that these key definitions are arranged in a simple Basic program, and that keys 1 and 2 are used to MOUNT and CATalogue a disc respectively. The command \*MOUNT is normally used to tell the ADFS that a new disc has been inserted, but it can also be used to get you back to the root directory of your current disc. The definitions of keys 3 and 4 are terminated with a \*CAT command, so that as well as selecting a given directory, its contents are automatically displayed. Just CHAIN the program to set up the keys.

To automate the process somewhat, why not create a !BOOT file to call the keys program every time that you press Shift-Break (i.e. hold down the Shift key, and at the same time tap the Break key)? A suitable !BOOT file can be generated as follows. First select the root directory of your disc (use \*DIR \$), then type:

\*BUILD !BOOT  
At the prompt, type:

CHAIN "\$.Keys"

Then press Return, then Escape. If you now type:

\*OPT 4,3

this will tell the computer to execute the !BOOT file as a command file whenever you perform a Shift-Break. If you now press Shift-Break, it should all happen, assuming of course that the Basic program containing your key definitions is called "Keys", and resides in the root directory.

```

10 REM FUNCTION KEY DEFINITIONS
20 *KEY1 *MOUNT|M*CAT|M
30 *KEY2 *CAT|M
40 *KEY3 *DIR $.PROGRAMS|M*CAT|M
50 *KEY4 *DIR $.GRAPHICS|M*CAT|M
60 etc
  
```

Figure 4  
Example function key definitions

*There will be more on using discs next month. In the mean time, if you have any queries, problems or tips, please drop us a line.*

RU

## SAFE FILE DELETION HINT

You can use \*SET ALIAS to create the new command \*DEL to give you a wildcarded file delete facility without the danger of wiping all files if you forget to add the C parameter required with \*WIPE. Use

\*SET ALIAS\$DEL WIPE %0\* C %1 %2 %3

Now if you execute \*DEL Prog the effect will be the same as executing \*WIPE Prog\* C. And \*DEL with no filename will offer all items in the current directory for possible deletion.

L.G.C

# VISUAL EFFECTS

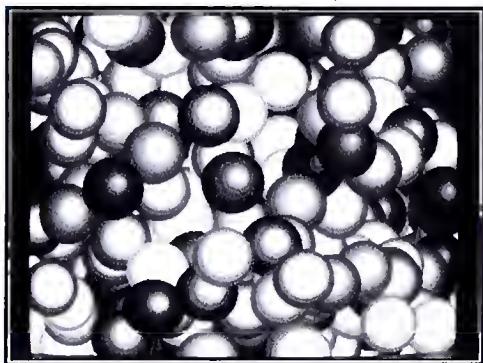
The Archimedes boasts many powerful graphics features, and can be used to generate quite stunning visual effects from very small amounts of code. We shall be using these columns in the coming months to give some idea of the machine's potential. This month's contribution is most appropriately from Paul Fellows at Acorn.

A305 SC 20 SP RMA -

## RANDOM SPHERES

The program *Spheres* uses a simple technique to create randomly coloured spheres and to distribute them randomly about the screen. The picture below gives little idea of the colourful effect created, and no idea of the relative speed with which the spheres are generated.

There are a number of tricks used in the program, but nothing too esoteric. The most important part is the generation of the spheres themselves. This is carried out in the procedure *PROCsphere*, which creates a single coloured sphere each time it is called. The spheres actually consist of a set of eight filled circles of progressively increasing lightness of shade and decreasing size. This is handled in the FOR-NEXT loop at line 230. To add to the 3 dimensional effect, the centre of each sphere is progressively offset upwards and to the right by a small amount (line 260).



```
10 REM >Spheres5
20 REM Program Random Spheres
30 REM Version A 0.5
40 REM Author Paul Fellows
50 REM Risc User November 1987
```

```
60 REM Copyright Acorn Computers Ltd
70 :
80 MODE15:OFF
90 step%:=8:radius%:=64
100 rad2%:=radius%>>1
110 L%:=LOG(512/radius%)/LOG2
120 REPEAT
130 VDU29,FMX;FNY;
140 PROCsphere
150 UNTIL FALSE
160 :
170 DEFFNX=RND(640)<<1
180 DEFFNY=RND(256)<<2
190 DEFFNrgb=(RND(3)-1)*1+(RND(3)-1)*4
+ (RND(3)-1)*16
200 :
210 DEFPROCsphere
220 Base%=FNrgb:GCOL Base%
230 FORX%:=radius% TO step% STEP=step%
240 VDU23,17,2,512-(X%<L%)|
250 IFX%<=rad2% GCOLBase%+&15
260 MOVE-X%DIV3,-X%DIV3:PLT&9D,X%,0
270 NEXT:ENDPROC
```

The colour of the spheres is determined at three different lines in the program. The base colour is selected randomly at line 220 by calling the function *FNrgb*. Line 240 cycles through 4 progressively lighter tints of the base colour, and the effect of this is doubled up (so producing 8 shades in total) by adding &15 to the base colour once the first 4 circles have been drawn (line 250).

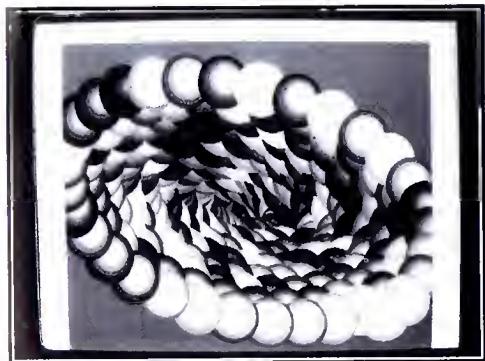
## LISSAJOU'S SPHERES

The second program listed here is based heavily on the first. It uses a slightly modified version of *PROCsphere*, and instead of placing the spheres randomly on the screen, they are made to follow a waveform made up of a randomly selected sin-cos combination. This results in the creation of a wide variety of interesting three dimensional images. To add to the visual effect, a randomly chosen background colour is used with each new pattern.

```

10 REM >Lissajou4
20 REM Program Lissajou's Spheres
30 REM Version A 0.4
40 REM Author Lee Calcraft
50 REM Risc User November 1987
60 REM Program Subject to Copyright
70 :
80 MODE15:OFF:VDU19,0,24,190,190,190
90 REPEAT

```



```

100 A=0:Q%=RND(60):R%=RND(10)
110 S=RND(4):T=RND(2):U=RND(3)
120 GCOL(RND(128)+128):CLG
130 REPEAT
140 A+=0.2:VDU29,FNX;FNY;
150 PROCsphere
160 UNTIL A>70
170 TIME=0:REPEAT UNTIL TIME=200
180 UNTIL FALSE
190 :
200 DEFFNX=640+(Q%-A)*10*COS(T*A+S)
210 DEFFNY=512+A*R%*SIN(A-U)
220 DEFFNrgb=(RND(3)-1)*1+(RND(3)-1)*4
+ (RND(3)-1)*16
230 :
240 DEFPROCsphere
250 Base%=FNRgb:GCOL Base%
260 FORX%=64 TO 8 STEP -8
270 VDU23,17,2,512-(X%<<3)|
280 IFX%=>32 GCOL Base%+15
290 MOVE-X%DIV3,-X%DIV3:PLOT&9D,X%,0
300 NEXT:ENDPROC

```

### ETERNAL BOUNCE

The third listing is again based on the sphere drawing procedure, though this time it has been adjusted to draw half size spheres. When the program is run you will see a single green sphere bouncing effortlessly around a

bordered screen. The program repeatedly applies the very powerful graphics block move command to the small rectangle containing the sphere (more on this in the next issue). This is moved around the screen in such a way as to simulate the trajectory of a ball with a restitution coefficient of one. The WAIT statement in line 180 ensures that the move is synchronised with the frame scan of the VDU, and ensures a very smooth gliding action. If you wish to change the speed of the ball, you can experiment with the values of F and G in line 100.

```

10 REM >Bounce5
20 REM Program Bouncing Sphere
30 REM Version A 0.5
40 REM Author Lee Calcraft
50 REM Risc User November 1987
60 REM Program Subject to Copyright
70 :
80 MODE15:OFF:VDU19,0,24,190,160,240
90 GCOL 160:CLG:Base%=8:GCOL Base%
100 X1=0:Y1=0:X=8:Y=800:F=8:G=-16
110 VDU29,68,68,:PROCsphere
120 VDU29,0,0,:MOVE 0,0:MOVE 110,110
130 REPEAT
140 IF X<8 OR X>1165 F=-F
150 IF Y<16 OR Y>900 G=-G
160 X+=F:Y+=G
170 MOVE X1,Y1:MOVE X1+110,Y1+110
180 WAIT:PLOT 189,X,Y:X1=X:Y1=Y
190 UNTIL FALSE
200 :
210 DEFPROCsphere
220 FORX%=-32 TO 4 STEP -4
230 VDU23,17,2,512-(X%<<4)|
240 IF X%<=16 GCOL Base%+15
250 MOVE-X%DIV2,-X%DIV2:PLOT &9D,X%,0
260 NEXT:ENDPROC

```

RU

### SEND US YOUR VISUALS

We are running a competition for the best Archimedes visuals. Winners will receive a £20 retail token, and will have their programs featured in these columns. So send us your contributions without delay!

Send to: Archimedes Visuals,  
RISC USER, Dolphin Place,  
Holywell Hill,  
St Albans, Herts, AL1 1EX

# USING THE EDITOR AND THE EMULATOR

David Graham explains how to make use of these two useful items from the Welcome Disc.

## THE BASIC EDITOR

The Welcome disc contains an excellent Basic Editor, described in some detail in the Basic User Guide. To get this up and running on your own disc, use the following procedure:

First create a directory for the module on your own disc. We will use a directory called MODULES:

```
*CDIR $MODULES
```

Next, make a copy of the module:

```
*COPY :0.$MODULES.ARMBE :0.$MODULES.P
```

Follow the machine's prompts (as summoned by the P parameter) to insert source and destination discs. It is assumed here that only a single drive is available for copying.

Now mount your own disc:

```
*MOUNT
```

and type:

```
*$MODULES.ARMBE
```

If you get the message *No room in RMA*, type QUIT, and perform the load again. Then use \*BASIC to get back to Basic. 305 users may also need to configure more RMA space - see Hints & Tips.

Now whenever you wish to use the editor, just type:

```
EDIT
```

from within Basic, and any program resident in the machine will be loaded into the editor. And don't forget to use the special keystrip supplied for the purpose.

## THE EMULATOR

The Welcome disc also contains a so-called 6502 Emulator. This is a piece of software which will allow you to run all 'legally written' programs from previous versions of the BBC micro. The programs may be in Basic or even in 6502 machine code. For example, you may

run disc-based versions of View directly on the Emulator.

To get the 6502 Emulator up and running from your own disc, copy the file called 65ARTHUR from the \$.MODULES directory of your Welcome disc, just as described above for the Editor:

```
*COPY :0.$MODULES.65ARTHUR :0.$MODULES.65ARTHUR P
```

To engage the Emulator, just type:

```
*$MODULES.65ARTHUR
```

You will now be in Basic VI. To verify this, type:

```
HELP
```

or QUIT

You will get the response:

```
Mistake
```

since although Archimedes Basic recognises these keywords, Basic VI (that used in the Master and Compact) does not. You may now run any legal program from earlier versions of the BBC micro - even those using 6502 assembler. But remember, any direct access to memory will probably crash the system.

If you have a RAM based version of View, such as that supplied with the Compact, this will work directly on the Emulator. To achieve this, type the following:

```
*$MODULES.65ARTHUR  
to get into the Emulator. Then
```

```
*RUN VIEW 8000
```

This assumes that View is in your current directory.

```
*$MODULES.65ARTHUR  
6502 Emulator 0.82 (01 Jun 1987)  
Acorn 6502 TOS 0.40 (11 Jun 1987)  
BASIC
```

Loading the emulator

# AN ALIAS FOR ARTHUR

by Lee Calcraft

The many variants of the operating system commands \*SET and \*SET ALIAS provide the user with massive power and flexibility in customising his Archimedes. Here are some brief notes based on our initial investigations.

## SET ALIAS

At the simplest level, you can use this command to rename already existing command sequences. For example:

```
*SET ALIAS$BEEB $.MODULES.65ARTHUR
```

will set up the new command \*BEEB. When this is called, Arthur will execute \*\$.MODULES.65ARTHUR. This will engage the 6502 Emulator (assuming that it resides in \$.MODULES). To take another example:

```
*SET ALIAS$TEXT DIR $.LETTERS.1987TEXT
```

will create the new command \*TEXT which will select directory 1987TEXT, a sub-directory of \$.LETTERS. Commands may also be linked using |M as a separator. Thus:

```
*SET ALIAS$M MOUNT|MCAT
```

will mount a disc and catalogue the root directory every time the command \*M is issued.

## ECHO

By incorporating the keyword ECHO, the alias can echo words and even VDU codes to the screen. For example:

```
*SET ALIAS$BORDER ECHO |<19>|<0>|<24>|<0>|<240>|<240>
```

will create the new command \*BORDER which sets up a cyan screen border. Each time the star command is issued, the operating system echoes the string of VDU characters 19,0,24,0,240,240.

## PARAMETERS

Parameters may be passed across into the alias string. As a trivial example:

```
*SET ALIAS$LOAD ECHO I won't load %0
```

will print the response:

I won't load filename  
every time that you try to execute:

```
*LOAD filename
```

The %0 indicates that the parameter given when the command is issued is to be incorporated at this point in the new command. You can in fact incorporate up to 9 parameters in any new command. Use %0 for the first, %1 for the second, and so on. Alternatively, the

sequence %\*0 sends the rest of the command line as a single parameter.

## SETTING FILE TYPE

File types may be changed with a further variant of the \*SET command. For example:

```
*SETTYPE filename FFE
```

will set the specified file to file type &FFE. This is the file type for command files, and will ensure that if you execute \*filename, the file will be EXECed in, and so will execute properly.

### Some Useful File Types

&FFF Plain ASCII	&FF9 Sprite
&FFE Command	&FF8 To run at &8000
&FFD Data	&FF7 BBC Font
&FFB Basic	&FF6 Fancy Font
&FFA Reloc Module	

## RUN TYPE

There is also a very handy command to set the course of action followed by the operating system when any file of a given type is called. For example:

```
*SET ALIAS$@RUNTYPE_FFD EXEC %*0
```

tells the operating system to EXEC in any file of type &FFD which is called using \*filename. This now means that any data files, if called in this way, will be EXECed in, without the need to use the \*SETTYPE command on every file.

## RUN PATH

You can also specify the directory paths which the operating system should search through in its attempts to find a given file. For example:

```
*SET RUN$PATH ,,$.MODULES
```

means that when you execute \*filename, Arthur will first check in the current directory (the space before the first comma indicates *current directory*), then it will look in the library directory (the % indicates this), and finally it will look in \$.MODULES. As you can appreciate,

## AN ALIAS FOR ARTHUR

this is a very useful command to set up, since it means, for example, that you can keep all your relocatable modules in a separate directory (e.g. \$.MODULES), but you can call any module with just:

\*modulename

without having to bother about directories. Of course you only need to call these various SET commands at power up and after Ctrl-Break; and they could all be put into a !BOOT file for automatic installation.

## ARTHURIAN VARIABLES

We now move on to further subtleties of Arthur. Acorn are beginning to build up a language in which the user may dialogue with Arthur, and it looks as if they will be extending these ideas in future releases. On Arthur 0.2 there are a number of Basic-like statements which can be used directly from the system prompt. For example, if you get into Arthur, and then type:

```
IF SYS$TIME LEFT 2>11 THEN ECHO Good  
Afternoon ELSE ECHO Good Morning
```

The result, as you might guess is that Arthur will say "Good Morning" or "Good Afternoon" as appropriate. Amongst other things here, we have made use of the "special" variable SYS\$TIME. You may also like to experiment with SYS\$DATE and SYS\$YEAR.

CLI\$PROMPT is another special variable, and is used to define Arthur's prompt. It is currently a star, but may be changed using \*SET. Thus:

```
*SET CLI$PROMPT Arthur>*
```

will give a prompt of:

Arthur>\*

any time that you are in supervisor mode. Perhaps more interestingly, you can make the prompt into the current time. For this you need a variant of the \*SET command which re-evaluates the assigned variable every time that it is called, thus:

```
*SETMACRO CLI$PROMPT <SYS$TIME>
```

In fact these SYS\$ and CLI\$ variables are a special case of a new breed of operating system variable. Try for example:

```
*SET Name FRED
```

If you now type \*ECHO Name, the result will be "Name", but if you use:

```
*ECHO <Name>
```

you will get the desired result "Fred".

This is all extremely powerful stuff - of which more later. In the mean time, I leave you with just one more command:

```
*UNSET name
```

It can take wild cards, and will delete previously set variables. Use \*SHOW to display all settings.

RU

## ARCHIMEDES DISC MENU (continued from page 9)

```
2740 EQUS STRING$(80,CHR$32):EQUB31  
2750 EQUB3:EQUB5  
2760 EQUS STRING$(74,CHR$32):EQUB31  
2770 EQUB3:EQUB6  
2780 EQUS STRING$(74,CHR$32):EQUB31  
2790 EQUB0:EQUB0:EQUB11  
2800 EQUS STRING$(80,CHR$32)  
2810 EQUB31:EQUB3:EQUB0  
2820 EQUS"Archimedes Disc Menu  
** RISC USER **"  
2830 EQUB17:EQUB0:EQUB17:EQUB128+8  
2840 EQUB31:EQUB63:EQUB5  
2850 EQUS"      QUIT      ":EQUB31:EQUB3  
2860 EQUB5:EQUS"      DIR ^      ":EQUB23  
2870 EQUB1:EQUD0:EQUD0:EQUS"$":ALIGN  
2880 .buff  
2890 ]:NEXT  
2900 OSCLI("SAVE RMENU "+STR$~code+" "+  
STR$~(buff+$1000))  
2910 *SETTYPE RMENU FFA  
2920 *RMLOAD RMENU  
2930 END  
2940:  
2950 DEFPROCsetup  
2960 wr=256:by=6:wo=7:rd=4  
2970 cli=5:ws=1:wc=0:ms=&1C  
2980 gb=&C:rs=&E:os=&16:es=&2C  
2990 ENDPROC
```

RU



# OFF THE SHELF



Mike Williams and Lee Calcraft have been looking at some of the first products available off the shelf for your Archimedes.

## DELTABASE

### MINERVA SYSTEMS, £29.95 inc VRT

Deltabase from database specialists Minerva is a sophisticated form of card index, and derives from Minerva's earlier System Delta and Card Index for the old BBC Micro and later Master series. Within the card index format Deltabase offers a wealth of facilities for entering, processing and extracting information in a card index file.

Over 8000 entries are possible in a single card index, with up to 255 fields per card. Card layouts are defined by the user and allow prompts, or other descriptive text, and entry fields to be laid out as required. You can also specify the order in which fields are to be entered, and fields may be linked together (for example the day, month and year).

Once a file and record format have been defined you can begin to enter data. At any time you may browse through the card records, search for particular entries (and if your knowledge of Basic is good, quite complex searches are possible), and display or print selected fields from each record. Card index files may be sorted using any combination of fields or part-fields to form a search key. Again, a knowledge of Basic will help in constructing complex sort keys which are evaluated using Basic's EVAL function.



Suffice it to say that for a card index the facilities are very comprehensive. I do, however, have two criticisms of Deltabase. The system is very strongly menu driven using the keyboard to select any option, and many Control-key combinations are also used. This is disappointing on a machine which has a mouse and all the supporting WIMPS software as standard. From the screen appearance Deltabase might almost be running on an ordinary BBC model B. Secondly, I find the manual poorly laid out, and difficult to read. I believe the user of a package such as this requires a well written, clearly laid-out introductory and tutorial section, followed by a concise and efficient reference to all the facilities. Minerva appears not to agree.

Deltabase certainly provides a more than adequate package for many database requirements based on the card index approach, but do be prepared for some difficult reading in almost every sense if you are going to make the most of what Deltabase has to offer.



In a short review it is impossible to cover all the facilities of a package such as Deltabase in detail.

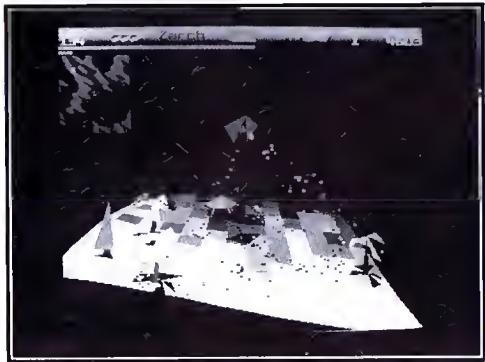
## ZARCH

### SUPERIOR SOFTWARE, £19.95 inc VRT

Everyone buying an Archimedes gets a taste of Zarch in the form of Lander, and if past experience is anything to go by, that alone has even the most humble games player drooling at the mouth over the super fast and impressive graphics. Zarch (what an odd name) is the real thing, and very good indeed it is too.

# OFF THE SHELF

The scenario of this game does sound somewhat contrived, as you seek out and destroy the environment-polluting 'Seekers', but in the end it boils down to a typical shoot-the-alien scene. What sets this game apart is the sheer speed of everything, from the scrolling multi-coloured landscape through to the desperately fought out aerial dogfights high in the sky.



On-screen displays show your altitude and fuel level throughout the game, and a radar scan (in large or small scale) is an essential adjunct for locating the enemy (and your home base for refuelling and re-arming). Beware of shooting the rotating radar towers though, or you will find your radar display partly blacked out.

Whether the game is worth the price asked for it you must judge for yourself, but if you've just bought an Archimedes, or are about to do so, then you might miss out on one of THE games of 1987. Buy now and enjoy the fun. M.C.W.

M.C.W.

TWIN

**ACORN SOFTWARE, £33.35 inc VAT**

Twin is a sophisticated multi-purpose text editor, and for those with experience of the BBC Master 128, it is very similar in operation to the Master Editor bundled with that machine. It differs from the Basic Editor supplied with the Archimedes in that it is not line number orientated. While the Basic Editor supplies line numbers, Twin offers a completely open format, and when it is used to edit a Basic program, it treats it just like any ASCII file. Having said that, ARM Basic recognises the keyword TWIN (try also TWINO 8), and in

response, will cause Twin to be loaded from disc with any resident Basic program suitably detokenised, and installed in Twin's text buffer

Twin is much more flexible than the ARM Basic Editor, and you can use it to examine and edit data files of all kinds. It is particularly useful for creating and editing command files including !BOOT files, which is not possible with the Basic Editor. It also has a much more sophisticated search and replace facility.

Finally, Twin boasts a powerful dual window mode from which, in fact, its name is derived. Using this facility, any two files may be concurrently displayed and edited. Data may be transferred between the windows, and there is even a measure of multi-tasking built in. You may for example compile and run the source code of a language such as C, while simultaneously using the other window to edit an object file.

## TOOLKIT

**CLARES, £39.95 inc VAT**

Clares Micro Supplies have developed a Toolkit for the Archimedes, and although not completely finished at the time of writing, it is due for release within a few days. Toolkit performs a number of useful disc and memory based activities. Many of these make use of an extremely fast full screen editor. In disc mode, this allows you to read and edit any byte on a disc, while in memory mode you can display and edit any byte of RAM. The editor can operate in a number of different formats including ARM instruction format, and Clares have even included a line by line assembly facility, allowing

# OFF THE SHELF

you to enter ARM assembler instructions directly at the keyboard. To assist the machine code programmer there are also a number of search operations, a block compare, and a memory move routine

Generally speaking this product is for the advanced user, though there are one or two useful options which will appeal to all, such as \*CATALL which will catalogue a whole disc, and \*DIRALL which will list all the directories on a disc

#### GENERAL:

*BIT	*ENVIRONMENT	*HCOMPARE
*HEX	*HFIND	*MEDIT
*MFIND	*MNEMONIC	*SHIFT
*SWAP	*TEXT	*TFIND
*WCOMPARE	*WFIND	*WORD

#### ADFS:

*AEDIT	*AHFIND	*AMFIND
*ATFIND	*AWFIND	*CATALL
*DIRALL	*EXALL	

#### Toolkit Commands

## ACORNSOFT C, PASCAL AND FORTRAN, £113.85 EACH inc VRT

Acornsoft's C is a full implementation of the draft ANSI standard, and is currently available in what Acorn call a pre-production package. Unlike their BBC micro C, recently released, this version does produce fully stand-alone code. When the full release is available, users may upgrade free of charge, but at present the documentation is very brief - just 55 pages, so you would be advised to wait until the full package becomes available unless you are already familiar with the C language.

Acornsoft Pascal and Fortran are also available off the shelf for the Archimedes. Again these are in a pre-production form, with free upgrades when the final versions become available.

#### WORDWISE PLUS

## COMPUTER CONCEPTS, £33.35 inc VRT

Wordwise Plus is now available on disc for the Archimedes. It runs under the Emulator only, and is a little on the slow side. But it does the job, and if

you are used to Wordwise, and wish to bridge the gap before a dedicated Archimedes word processor becomes available, the product (as an upgrade) is very reasonably priced at £11.50.

## ARM ASSEMBLY LANGUAGE, MTC, £12.95 inc VRT

This very useful book written by Peter Cockerell, and available from the Computer Concepts stable has been on the shelves for a number of months now. Indeed it was available before the machine itself. It is very competently written, and extremely easy to follow. My only criticism is that it could contain more examples in the early stages of the book. But this book is really a must for anyone interested in ARM assembler.

## ARCHIMEDES PROGRAMMER'S REFERENCE MANUAL, RCORN, £29.95 inc VRT

This important two-part tome should be available by the time you read these words. It offers quite extensive documentation on the machine, and will prove invaluable to anyone who has moved out of the beginner's league. Unfortunately there is a large gap between the level covered in this guide, and that of the User Guide. The Programmer's Manual is crammed with information, but much of it is too briefly expressed for those new to the machine. Obviously, with a machine as complex as the Archimedes, more documentation is badly needed, especially works which can bridge the gap between the technical subtleties of the Programmer's Reference Manual and the Basic User Guide. But Acorn have done well to get this product into the shops at this stage.

L.G.C.

Minerva Systems,  
69 sidwell St, Exeter,  
Devon EX4 6PH.

Clares Micro Supplies,  
98 Middlewich Road,  
Rudheath, Northwich,  
Cheshire CW9 7DA.

Acorn Computers Ltd.,  
Cambridge Technopark,  
645 Newmarket Rd.,  
Cambridge CB5 8PD.

Computer Concepts,  
Gaddesden Place,  
Hemel Hempstead HP2 6EX.

Superior Software,  
Regent House,  
Skinner Lane,  
Leeds LS7 1AX.

RU

# HINTS & TIPS HINTS & TIPS

In our first issue of RISC User, we bring you a special double page crop of hints and tips, supplied by Rob Barnes, Graham Stanley and Lee Calcraft.

## BACKING UP

To make a backup copy of a disc, Acorn advise that you use the special program Newbackup found in the Library directory of your Welcome disc. With your Welcome disc in drive zero, type:

\*\$.LIBRARY.Newbackup

Then follow the prompts.

## \*COPY OPTIONS

Use \*HELP COPY to find out about all the possible options for use with \*COPY. Beware of the Q or Quick option, though. This sometimes caused our machines to hang. And remember that you must always give a destination filename when using \*COPY (not necessary with the ADFS in earlier BBC micros).

## WIPE WARNING

The excellent command \*WIPE allows you to selectively delete files to clean up your discs. Use \*HELP WIPE to find out the various options. But be warned that:

\*WIPE \*

will wipe all files from your current directory without any warnings! Worse still:

\*WIPE \* FR

will completely wipe the whole contents of a disc, scouring out even the deepest nested directories, and completely ignoring the locked status of any files. Again, no prompts are given.

## STAR SPACE

When communicating with Arthur using star commands which require parameters, it is essential to leave a space before the first parameter. For example, if you use:

\*MOUNTO

instead of \*MOUNT O you will get a "Bad command" message.

## \*HELP

The commands:

\*HELP

and \*HELP topic

give masses of useful information. To display all the machine's help messages, use:

\*HELP .

The space before the full stop is essential. Don't forget to press Shift to scroll the display. You may find it worth printing this out by pressing Ctrl-B before issuing the Help command.

## SCROLL VARIANTS

When using the LIST command to list a Basic program, the following may be useful:

Press Ctrl and Shift to freeze

Press Scroll-Lock to freeze/unfreeze

Press Shift to slow the scroll

Additionally using Ctrl-N before listing will page the display. Then use Shift to scroll, and Ctrl-O to cancel the effect.

## LANDER HINT

When playing the Welcome Lander game, the following may be useful:

Scroll-Lock will pause the game

Ctrl will slow it down

## PRINTER LEAD

The Archimedes parallel printer socket is just like that used on the IBM PC. This means that a standard IBM printer lead can be used. But you may find that BBC stockists can supply this item more cheaply than business computing suppliers.

## STANDARD STAND

You may not have spotted that both the keyboard and monitor have built-in stands which fold out from underneath. The keyboard stand tilts the keyboard to a very convenient angle.

## LIST IF

The command

LIST IF word

should find all occurrences of the word *word* in your program. But it will not work properly unless you leave NO space between the IF and the word. So, for example, to find all procedures, use:

LIST IFPROC

## QUITTING BASIC

When loading modules of various kinds, you may have received the message *No room in RMA* (RMA=Relocatable Module Area). You can often get around the problem by quitting Basic - use:

QUIT

Next, perform your module load. Then re-enter Basic by typing:

BASIC

As you will notice, Arthur provides the asterisk.

# HINTS & TIPS HINTS & TIPS

## LIST VARIABLES & LIBRARIES

The very useful command:

LVAR

will list all variables and arrays, including the resident integer variables. It also gives a list of all resident Basic libraries.

## BASIC HELP

The command:

HELP

on its own (i.e. without the asterisk) gives useful Basic status information, including program length and the amount of free memory. Try also the extensive HELP facility in RAM Basic.

## SINGLE STEPPING

Use:

TRACE ON

to display line numbers as a program runs. To single-step through the program, ignore the User Guide, and type:

TRACE 'STEP ON

Try it with the Spheres programs in this issue. Type RUN, then press the space bar to advance each step.

## BORDERING

To generate coloured screen borders outside the normal graphics area, use:

VDU19, 0, 24, r\*16, g\*16, b\*16

where r, g and b can take values from 0 to 15, specifying the red, green and blue components of the border. For example:

VDU19, 0, 24, 0, 240, 240

gives a cyan border (max green and blue with no red).

## CURSOR ON/OFF

There are two very simple commands to turn the screen cursor on and off. They are:

ON

and OFF

which should be pretty easy to remember!

## AUTO-SAVE

If you include a REM line as the first line of your program in the following way:

10 REM >filename

every time that you type:

SAVE

giving no filename, Basic will save your program using the name given in the REM statement. To make the operation even slicker, try programming a function key to do the work for you:

\*KEY0 SAVE|M

will cause a save to be made every time that the Print key (which doubles as function key zero) is pressed.

## INVERSE VIDEO

The command:

VDU23, 17, 5|

will reverse the video output. This gives black text on a white background, which can be much easier to read than white on black. To return things to normal, either issue the command again, or execute a mode change.

## PROGRAM END

Use:

PRINT END

to give the end of a program in memory, taking variable storage into account.

## COMMAND FILE TYPING

If you create a command file using \*BUILD, or a word processor such as View, Wordwise or the text editor Twin, you will find that although you can use:

\*EXEC

to call it, you cannot just give its name to the operating system to EXEC it in. You will get the error message *Code runs too low*. To make Arthur realise that it is a command file, you need to change the file's file type. You achieve this as follows:

\*SETTYPE filename FFE

where *filename* is the name of the command file. The letters FFE represent the hex file type for a command file.

## CONFIGURING 305 SYSTEMS

It is often possible to reconfigure Archimedes 305 machines to run programs which would normally require a 310. For example, to configure a 305 to cope with screen mode 15 (and avoid the Bad mode error), use:

\*CON. SCREENSIZE 20

Then use Ctrl-Break (press the Ctrl key and the Break key simultaneously). Use \*CON. SCREENSIZE 0 to set it back again.

Alternatively, if you get the message *No room in RMA* then try reconfiguring the RMA size with:

\*CON. RMASIZE n

If n=1 this reserves 8K, n=2 gives 16K, and so on. Obviously you should reserve no more than needed by a particular application when using a 305. And again, as with all \*CONFIGURE options, you must press Ctrl-Break before it takes effect. To give an example, you need to set n=15 to run the 6502 Emulator or the Basic Editor.

RU

# RISC USER magazine

## MEMBERSHIP

RISC User is available only on subscription, with a special introductory rate until early 1988. Full subscribers to RISC User may also take out a reduced rate subscription to BEEBUG (the magazine for the BBC micro and Master series).

All subscriptions, including overseas, should be in pounds sterling. We will also accept payment by Connect, Access and Visa, and official UK orders are welcome.

### BACK ISSUES

We intend to maintain stocks of back issues as they become available. New subscribers can therefore obtain earlier copies to provide a complete set from Vol. 1 Issue 1. Back issues cost £1.20 each. You should also include postage as shown:

Destination	UK, BFPO, Ch.Is	Europe plus Eire	Elsewhere
First Issue	40p	75p	£2
Each subsequent issue	20p	45p	85p

### Subscription Rates (12 Months)

	RISC User	BEEBUG (reduced rates)
UK, BFPO, Ch.Is	£12.50	£ 8.00
Rest of Europe & Eire	£18.00	.. £10.00
Middle East	£22.50	£11.00
Americas & Africa	£25.00	£11.50
Elsewhere	£27.00	£12.00

## MAGAZINE DISC

It is our intention to publish each month all the programs from that issue of RISC User on 3.5" disc. This will be available to order, or you may take out a subscription to ensure that the disc arrives at the same time as the magazine. The first issue of RISC User contains six programs and these are available at the special low price of £3.75.

The prices for future issues are shown below:

	UK	Overseas
Single issue discs	£ 4.75	£ 4.75
Six months subscription	£25.50	£30.00
Twelve months subscription	£50.00	£56.00

Disc subscriptions include postage, but you should add 50p per disc for individual orders.

All orders, subscriptions and other correspondence should be addressed to:

**RISC User**, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX.

Telephone: St Albans (0727) 40303

(24hrs answerphone service for payment by Connect, Access or Visa card)